Manifold Learning

Máster Universitario en Ciencia de Datos - Métodos Funcionales en Aprendizaje Automático

Ángela Fernández Pascual

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Academic Year 2021/22

Universidad Autónoma
de Madrid

Outcomes

By the end of this session you will...

1. be able to distinguish manifold learning algorithms
2. have experimented with different manifold learning methods
3. argue which one is the best one for a particular problem

Contents

Manifold Learning Techniques

Manifold Learning

- **Non-Linear** Dimensionality Reduction methods.
- **Spectral** methods: spectral analysis of a similarity matrix.
- Objectives:
  - To describe properly the **geometry** of the data.
  - To find a good **distance** in the original space equivalent to a Euclidean distance in the embedding.
  - To consider when patterns $\mathbf{x}^{(i)}$ are in a differentiable **manifold** $\mathscr{M} \subset \mathbb{R}^M$ of dimension $\bar{M}$.

Some Manifold Learning Techniques

- Multidimensional Scaling (MDS)
- Isomap
- Kernel Principal Components Analysis (KPCA)
- Spectral Clustering (SC)
- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LE)
- Diffusion Maps (DM)

Multidimensional Scaling (MDS)

MDS Algorithm

1. Compute the Euclidean distance matrix $\mathbf{D}$.
2. Normalize the matrix as: $\mathbf{A} = \{-\frac{1}{2} d_{ij}^2\}$.
3. Compute the similarity matrix: $\mathbf{B} = \{a_{ij} - a_{i\cdot} - a_{j\cdot} - a_{\cdot\cdot}\}$, where $a_{i\cdot} = \frac{1}{N}\sum_k a_{ik}$
4. $\bar{M}$-dimensional Embedding: first $\bar{M}$ $\{\lambda_\ell, \vartheta_\ell\}$ of $\mathbf{B}$:

$$\boxed{\Phi : \mathbf{x}^{(i)} \rightarrow (\sqrt{\lambda_1}\vartheta_1(\mathbf{x}^{(i)}), \ldots, \sqrt{\lambda_{\bar{M}}}\vartheta_{\bar{M}}(\mathbf{x}^{(i)}))}.$$

```
class sklearn.manifold.MDS(n_components=2, *, metric=True, n_init=4, max_iter=300,
        verbose=0, eps=0.001, n_jobs=None, random_state=None, dissimilarity='euclidean')
```

Isomap

Isomap Algorithm

1. Select the neighbourhood: ϵ-distance or K-distance.
2. Compute the shortest paths in $\mathbf{D_G}$ (geodesic distance).
3. $\bar{M}$-dimensional Embedding: first $\bar{M}$ $\{\lambda_\ell, \vartheta_\ell\}$ of $\mathbf{D_G}$:

$$\Phi : \mathbf{x}^{(i)} \rightarrow (\sqrt{\lambda_1}\vartheta_1(\mathbf{x}^{(i)}), \ldots, \sqrt{\lambda_{\bar{M}}}\vartheta_{\bar{M}}(\mathbf{x}^{(i)})).$$

```
class sklearn.manifold.Isomap(*, n_neighbors=5, n_components=2, eigen_solver='auto', tol=0,
    max_iter=None, path_method='auto', neighbors_algorithm='auto', n_jobs=None,
    metric='minkowski', p=2, metric_params=None)
```

Notebook

MDS
Isomap

Kernel Principal Components Analysis (KPCA)

KPCA Algorithm

1. Define a Kernel matrix:

$$\mathbf{K} = \{k_{ij}\} = \left\{ e^{\frac{-\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\sigma^2}} \right\}.$$

2. $\bar{M}$-dimensional Embedding: first $\bar{M}$ $\{\lambda_\ell, \vartheta_\ell\}$ of $\mathbf{K}$, without taking into account λ_0 and ϑ_0:

$$\Phi : \mathbf{x}^{(i)} \rightarrow (\sqrt{\lambda_1}\vartheta_1(\mathbf{x}^{(i)}), \ldots, \sqrt{\lambda_{\bar{M}}}\vartheta_{\bar{M}}(\mathbf{x}^{(i)})).$$

```
class sklearn.decomposition.KernelPCA(n_components=None, *, kernel='linear', gamma=None,
    degree=3, coef0=1, kernel_params=None, alpha=1.0, fit_inverse_transform=False,
    eigen_solver='auto', tol=0, max_iter=None, remove_zero_eig=False, random_state=None,
    copy_X=True, n_jobs=None)
```

Spectral Clustering (SC)

SC Algorithm 1

1. Define a weighted graph using: ϵ-neighbourhood graph, KNN graph or fully connected graph.
2. $\bar{m}$-dimensional Embedding: first $\bar{m}$ $\{\lambda_\ell, \vartheta_\ell\}$ of $\boxed{\mathbf{L} = \mathbf{D} - \mathbf{W}}$, without taking into account ϑ_0:

$$\boxed{\vartheta : \mathbf{x}^{(i)} \rightarrow (\vartheta_1(\mathbf{x}^{(i)}), \ldots, \vartheta_{\bar{m}}(\mathbf{x}^{(i)}))}.$$

SC Algorithm 2

1. Define a weighted graph using: ϵ-neighbourhood graph, KNN graph or fully connected graph.
2. $\bar{m}$-dimensional Embedding: first $\bar{m}$ $\{\lambda_\ell, \psi_\ell\}$ of $\boxed{\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}}$, without taking into account ψ_0:

$$\boxed{\Psi : \mathbf{x}^{(i)} \rightarrow (\psi_1(\mathbf{x}^{(i)}), \ldots, \psi_{\bar{m}}(\mathbf{x}^{(i)}))}.$$

Spectral Clustering (SC)

SC Algorithm 3

1. Define a weighted graph using: ϵ-neighbourhood graph, KNN graph or fully connected graph.
2. $\bar{M}$-dimensional Embedding: first $\bar{M}$ $\{\lambda_\ell, \psi_\ell\}$ of $\boxed{\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}}$, without taking into account ϑ_0:

$$\boxed{\vartheta : \mathbf{x}^{(i)} \rightarrow \left(\frac{\vartheta_1(\mathbf{x}^{(i)})}{\sum_k \vartheta_k^2(\mathbf{x}^{(i)})}, \ldots, \frac{\vartheta_{\bar{M}}(\mathbf{x}^{(i)})}{\sum_k \vartheta_k^2(\mathbf{x}^{(i)})} \right)}.$$

```
class sklearn.cluster.SpectralClustering(n_clusters=8, *, eigen_solver=None,
    n_components=None, random_state=None, n_init=10, gamma=1.0, affinity='rbf',
    n_neighbors=10, eigen_tol=0.0, assign_labels='kmeans', degree=3, coef0=1,
    kernel_params=None, n_jobs=None, verbose=False)
```

Notebook

KPCA
SC

Locally Linear Embedding (LLE)

LLE Algorithm

1. Select the neighbourhood: ϵ-neighbourhood graph or KNN graph.
2. Find the optimal weights, solving:

$$\arg\min_{\mathbf{W}} \left\{ \sum_i (\mathbf{x}^{(i)} - \sum_j w_{ij} \mathbf{x_i}^{(j)})^2 \right\} \; s.t. \; \begin{cases} w_{ij} = 0 & \text{if } \mathbf{x}^{(j)} \notin \{\mathbf{x_i}^{(k)}\} \\ \sum_j w_{ij} = 1 & \forall i \\ w_{ij} > 0. \end{cases}$$

3. $\bar{M}$-dimensional Embedding: first $\bar{M}$ eigenvectors of $\mathbf{L} = \mathbf{D} - \mathbf{W}$, without taking into account ϑ_0:

$$\vartheta : \mathbf{x}^{(i)} \rightarrow (\vartheta_1(\mathbf{x}^{(i)}), \ldots, \vartheta_{\bar{M}}(\mathbf{x}^{(i)})).$$

Locally Linear Embedding (LLE)

```
class sklearn.manifold.LocallyLinearEmbedding(*, n_neighbors=5, n_components=2, reg=0.001,
    eigen_solver='auto', tol=1e-06, max_iter=100, method='standard', hessian_tol=0.0001,
    modified_tol=1e-12, neighbors_algorithm='auto', random_state=None, n_jobs=None)
```

Laplacian Eigenmaps (LE)

LE Algorithm

1. Define a weighted graph, with $\mathbf{W}$ defined by the Heat kernel:

$$w_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{4t}} & \text{if } (i,j) \text{ are connected,} \\ 0 & \text{if } (i,j) \text{ are not connected.} \end{cases}$$

2. $\bar{m}$-dimensional Embedding: first $\bar{m}$ eigenvectors of $\mathbf{L} = \mathbf{D} - \mathbf{W}$, without taking into account ϑ_0:

$$\vartheta : \mathbf{x}^{(i)} \longrightarrow (\vartheta_1(\mathbf{x}^{(i)}), \ldots, \vartheta_{\bar{m}}(\mathbf{x}^{(i)})).$$

```
class sklearn.manifold.SpectralEmbedding(n_components=2, *, affinity='nearest_neighbors',
    gamma=None, random_state=None, eigen_solver=None, n_neighbors=None, n_jobs=None)
```

Notebook

LLE
LE

Diffusion Maps (DM)

DM Algorithm

1. Define a weighted graph using a kernel:

$$\mathbf{K} = \{k_{ij}\} = \left\{ e^{\frac{-\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\sigma^2}} \right\}.$$

2. Define a random walk over the graph, via the transition probability:

$$p_{ij} = \left(\frac{k_{ij}}{d_i}\right) = \left(\frac{k_{ij}}{\sum_j k_{ij}}\right).$$

3. $\bar{M}$-dimensional Embedding: first $\bar{M}$ $\{\lambda_\ell, \psi_\ell\}$ of $\mathbf{L} = \mathbf{P} = \mathbf{D}^{-1}\mathbf{K}$, without taking into account ψ_0:

$$\Psi_T : \mathbf{x}^{(i)} \rightarrow (\lambda_1^T \psi_1(\mathbf{x}^{(i)}), \ldots, \lambda_{\bar{M}}^T \psi_{\bar{M}}(\mathbf{x}^{(i)})).$$

General ML Algorithm

General ML Algorithm

1. We start from a sample dataset $\mathscr{S} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$.
2. Scale features.
3. Compute the similarity matrix $\mathbf{W}$
   $\rightarrow$ normalize $\mathbf{W}$?
4. Spectral decomposition of $\mathbf{W}$ ($\{\lambda_\ell, \vartheta_\ell\}$).
   Select just the first $\bar{M}$ excluding the trivial solution.
5. New coordinates: a function of the eigenvectors.

Comparison between Manifold Learning Techniques

Method	Kernel Normalization	Density Normalization	Coordinates	Eigenvalue Order
MDS	$\mathbf{B} = \{a_{ij} - a_{i\cdot} - a_{j\cdot} - a_{\cdot\cdot}\}$	NO	$\sqrt{\lambda_\ell}\vartheta_\ell$	$\downarrow$
Isomap	$\mathbf{W}$	NO	$\sqrt{\lambda_\ell}\vartheta_\ell$	$\downarrow$
KPCA	$\mathbf{W}$	NO	$\sqrt{\lambda_\ell}\vartheta_\ell$	$\downarrow$
LE/LLE	$\mathbf{D} - \mathbf{W}$	NO	ϑ_ℓ	$\uparrow$
SC	$\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$	NO	ϑ_ℓ	$\uparrow$
DM	$\mathbf{D}^{-1}\mathbf{W}$	YES	$\lambda_\ell^\tau \psi_\ell$	$\downarrow$

Conclusions

Advantages

Advantages

- They search a new representation that captures the main structure of the original data.
- Non-linear methods.
- They are able to unfold a manifold.
- In general, they look for a **distance over the original sample** that is equivalent to the **Euclidean distance in the embedding**.

Disadvantages

Disadvantages

- **Out-of-sample points**
  - For new points the whole method should be repeated $\Rightarrow$ not feasible!
  - Possible solutions:
    - Methods that approach the eigenvectors for new points: Nyström.

- **Computational Cost**
  - Due to the eigendecomposition of the similarity matrix.
  - Possible solutions:
    - Low-rank approximation.
    - Searching fast eigenanalysis methods.

Summing up

You now should...

1. distinguish between different manifold learning algorithms
2. have experimented with different manifold learning methods
3. be able to argue which method is the best one for a particular problem

References

References

- Kruskal, J. B. (1964). **Multidimensional Scaling by optimizing goodness of fit to a nonmetric hypothesis**. Psychometrika, 29(1), 1-27.
- Tenenbaum, J. B., De Silva, V., Langford, J. C. (2000). **A global geometric framework for nonlinear dimensionality reduction**. Science, 290(5500), 2319-2323.
- Schölkopf, B., Smola, A., Müller, K. R. (1997). **Kernel Principal Component Analysis**. In International conference on artificial neural networks (pp. 583-588). Springer, Berlin, Heidelberg.
- Von Luxburg, U. (2007). **A tutorial on Spectral Clustering**. Statistics and computing, 17(4), 395-416.
- Ng, A. Y., Jordan, M. I., Weiss, Y. (2002). **On Spectral Clustering: Analysis and an algorithm**. Advances in neural information processing systems, 2, 849-856.
- Shi, J., Malik, J. (2000). **Normalized cuts and image segmentation**. IEEE Transactions on pattern analysis and machine intelligence, 22(8), 888-905.